

Erik Bartmann

Discover modular synthesizers with **VCV-Rack 2**



Foreword by

Omri Cohen

VCV-Rack-Book Topics / Themes

Chapter	Themes
1. Introduction	<ul style="list-style-type: none">• Non-modular and modular synthesizers• Signal flow (oscillator, filter, amplifier)• Modular hardware synthesizers• The four classic waveforms (sine, triangle, rectangle, sawtooth)• CV control signals• Patchkable
2. VCV Rack handling	<ul style="list-style-type: none">• What means VST• The installation of VCV-Rack 2• The start of VCV-Rack 2• The template• Add modules• Mark a module as Favorite• The empty VCV Rack• Move + position modules• Remove modules• Connecting modules to each other• Module input and output sockets• Color choice of the connection cables• Extended colors via plugin WM-101 from Submarine• Correct and remove connections• Multiple connections on one socket• Change visible section in VCV Rack (scroll, zoom, move)• Change patch cable parameters (opacity and tension)• The context menu of the modules• The installation of foreign plugins• Registration with VCV-Rack• The local authentication• The local plugin folder• The subscription to plugins• The cancellation of plugins• Regular updates
3. VCV Rack Introduction	<ul style="list-style-type: none">• The launch of VCV-Rack• The Start-Template• The terminology used• The communication between the modules• The signal flow• The input - The keyboard<ul style="list-style-type: none">○ What is a master keyboard○ The MIDI-CV-Module• What means MIDI• The Notes-Module• The output - The speakers<ul style="list-style-type: none">○ The Audio-8-Module• What means latency• The ASIO driver

	<ul style="list-style-type: none"> • What does sample rate mean • What means block size • The VCO module - The oscillator • What means octave and what are semitones • Notes and their voltage values • What means oscillation • What is a waveform • The VCA module - The amplifier?! • What is an amplifier • The VCA MIX module • What means modulation • What means velocity • The ADSR module - The envelope control • What is an envelope (curve) • The VCF module - The filtering • What are harmonics or overtones • What is a low-pass filter • What is a high pass filter • The summary of voltage levels (audio, CV)
4. Oscillators	<ul style="list-style-type: none"> • What is an oscillator • The VCO-Module • The difference between analog / digital waveform generation • The four basic waveforms (sine, triangle, sawtooth, square) • Harmonic spectrum of the four basic waveforms • Plugin provider Bogaudio • Inner socket colors at different voltage levels (negative, positive, AC, 0V) • What means Hertz • How are voltage and frequency related • The manual modulation of the frequency • What means Pulse Width Modulation - PWM • What does SYNC do at the VCO • What is cross modulation • The WT-VCO-Module • What is a wavetable • What means morphing
5. Sound synthesis	<ul style="list-style-type: none"> • What means sound synthesis <ul style="list-style-type: none"> ○ Additive sound synthesis ○ Subtractive sound synthesis • What are harmonics • Sound synthesis for a square wave signal • Sound synthesis for a sawtooth signal • Sound synthesis for a triangle signal • What means decibel • Other sound synthesis <ul style="list-style-type: none"> ○ Wavetable ○ FM ○ Cross-Modulation ○ PWM ○ Granular • Plugin provider Erica-Synth

6. LFO	<ul style="list-style-type: none"> • What is an LFO • The difference between unipolar and bipolar oscillation • The vibrato effect • The tremolo effect • The frequency modulation of the LFO • The pulse width modulation further explained • What does phase inversion of an oscillation mean • The PULSES-Module • The CLK input on the LFO
7. Filter	<ul style="list-style-type: none"> • What is a low-pass filter (LP) and what is a high-pass filter (HP) • What means damping / attenuation • What means slope • The VCF-Module • What is the cutoff frequency • What is resonance • What means self-oscillation • What means legato • Other types of filters <ul style="list-style-type: none"> ○ Bandpass-Filter (BP) ○ Notch-Filter
8. ADSR	<ul style="list-style-type: none"> • The ADSR-Module • What is an envelope curve • When is an envelope triggered • What does attack, delay, sustain and release mean? • The volume envelope (amp envelope) • The filter envelope • The pitch envelope • What causes a gate pulse that is too short
9. VCA	<ul style="list-style-type: none"> • The VCA-Module • The VCA-MIX-Module • What does envelope shaping mean • The tremolo effect once again explained • What is an attenuverter • What is a phase shift • The plugin provider Befaco
10. Mixing	<ul style="list-style-type: none"> • What means mixing • What means mastering • What is an equalizer • What is a compressor • Mixing with the VCA MIX module • The summation of signals • Invert a signal • Phase cancellation made visible • The 8VERT-Module • The MUTES-Module • Mixers from MindMeld • Equalizer from MindMeld
11. Frequency modulation	<ul style="list-style-type: none"> • What means frequency modulation • What is an algorithm • The interaction between modulator and carrier

	<ul style="list-style-type: none"> • What are sidebands • The establishment of feedback • Different algorithms • The MIX-Module
12. Sample & Hold	<ul style="list-style-type: none"> • What does Sample & Hold mean • The Kinks-Module (Audible Instruments) • Generate random values • The Clock-Module (JW-Modules) • The Delay-Module (used briefly) • The Sample&Hold-Modules of Count-Modula
13. Quantize	<ul style="list-style-type: none"> • What means quantize • The QNT-Module • What is the C major scale • What means harmonization • The LFO controls the VCO with the QNT • Two quantizers control tone sequence
14. Tone sequences	<ul style="list-style-type: none"> • What is a sequencer • The SEQ-3-Module • What are Steps • What is the difference between trigger and gate • The DGATE-Module from Bogaudio • What triggers a sequencer • The adjustment of the step length • The triggering of a reset • An external clock • What are patterns • The 4-1-Module • What does pattern chaining mean • Create your own waveforms • What does Portamento / Glide mean • The Befaco Slew Limiter Module • The RANDOM-Module • The GRID-SEQ module from JW-Modules
15. Adjust octaves	<ul style="list-style-type: none"> • The OCT-Module • Manual adjustment of the octave • Automatic adjustment of the octave • The 8VERT module - BIAS control • The Fixed Voltage Source Module
16. Delay	<ul style="list-style-type: none"> • The Delay-Module • How the Delay effect works • What is a decaying vibration • What does WET and DRY mean respectively • The Plateau Module from Valley
17. Noise	<ul style="list-style-type: none"> • What means noise • What does sound synthesis mean in terms of noise • What does decibel mean (subjective volume perception)? • The NOISE-Module • How to simulate ocean noise
18. Macro-Oscillator	<ul style="list-style-type: none"> • What are Macro Oscillators • What means wavefolding

	<ul style="list-style-type: none"> • What does waveshaping mean • The Braids and Plaits modules from Audible Instruments • What means soft clipping • The wavefolding module from MSM • The Wolv waveshaper module Vult • What means ring modulation • The Ring-Mmod module from Bogaudio • The Macro Oscillators Braids and Plaits • What is a clock divider • The GRID-SEQ module from JW-Modules
19. Polyphony	<ul style="list-style-type: none"> • What means Monophonic • What means Polyphonic • The different polyphonic modes • The VIZ-Module • The SPLIT-Module • What is a chord • Diphthong, triad, tetrachord • The C major triad • What is an arpeggiator • The Sequential-Switch module from the ML-Modules • The MERGE-Module • The MIX-4 module from Bogaudio
20. Drums	<ul style="list-style-type: none"> • What are percussive sounds • What is a bass drum • What are transients • What is a snare drum • The paid drum modules of VCV-Rack 2 • The Drum-Modules of Hora-Music • The Step-Trigger-Sequencer-Modul of Count-Modula • The Oscilloscope-Module of Count-Modula • The Drum-Kit of Autodafe
21. Sidechain	<ul style="list-style-type: none"> • What means sidechaining • What means ducking
22. Midi-Mapping	<ul style="list-style-type: none"> • What is MIDI-Mapping • What is a Byte or a Bit • The three bytes of a MIDI transmission • What are status byte, data byte 1 and data byte 2 • What does a MIDI Control Change (CC) mean • The MIDI-Monitor-Program MIDIView • The MIDI-MAP-Modul • Setting up a MIDI mapping • Solving a MIDI mapping
33. Logic-Modules	<ul style="list-style-type: none"> • What does binary logic mean • What is a binary code • What is Boolean Algebra • What is an AND function • What is an OR function • What is an exclusive OR function • What is a NOT function • The Triggers MKI module from AS

	<ul style="list-style-type: none"> • The BOOL-Modul from Bogaudio • The Logik-Modules from Submarine • The Logik-Module from NYSTHI
24. Sound-Input	<ul style="list-style-type: none"> • How to get external sounds into the VCV Rack • The configuration of the audio module • A microphone is connected • The use of an external audio interface • The REC-Module • The WT VCO loads a sound file • The Simplicer module from NYSTHI
25. Control hardware	<ul style="list-style-type: none"> • The control of hardware • The CV-to-MIDI-Module • The control of a real synthesizer • MIDI or USB • Recording an audio stream from an external audio interface
26. VST	<ul style="list-style-type: none"> • What is a DAW • What does VST mean for the DAW • The Installation of the VST • Adding a VCV VST to the DAW • The VST receives MIDI from the DAW • CV information from the DAW • The Clocked-Modul from Impromptu • The automation with the DAW • Multitrack recording in the DAW • What means routing • Send CV from VCV via MIDI
27. Performance	<ul style="list-style-type: none"> • VCV-Rack Performance • The Performance-Meter • The Block-Size • The Sample-Rate • The number of threads • The Frame-Rate
28. VCV-Plugins-Overview	<ul style="list-style-type: none"> • Interesting VCV Rack Plugin Brands

The VCV-Rack - The sound synthesis



What is it all about?

The following topics are discussed in this chapter.

- What means sound synthesis?
- What is additive sound synthesis?
- What is subtractive sound synthesis?
- What are harmonics?

As announced in the last chapter, we will now move on to sound synthesis. Here the already mentioned harmonics play a decisive role. There are two basic approaches to creating and shaping a sound or waveform the way you want it. The starting point is actually two waveforms that have a more or less wide spectrum in terms of harmonics. We can say, for example, that a generated sound has a certain fundamental tone and this is then enriched with more or less other frequencies, which makes up the overtones and makes this sound characteristic and unique. To generate sounds, there is a simple and a more complex way, which I would like to address at. There are many other methods besides the ones mentioned here, but they are not the subject of this chapter or book. So I will first start with the complex procedure and then switch to the simpler one. Because of the already mentioned problem with the standard VCO, which at the moment is not able to offer a digital procedure for the generation of the waveforms, I use a VCO of a third-party supplier. It's called Bogaudio and has already been mentioned as providing the spectrum analyzer. For safety here again the link.



Hyperlink!

<https://library.vcvrack.com/?brand=Bogaudio>

The additive sound synthesis

As the name suggests, additive sound synthesis involves adding something to a specific sound. The starting point is a certain waveform that only has a fundamental and no harmonics. We are talking here about the already known sine wave. Concerning the additive sound synthesis a well-known French researcher, who was a mathematician and physicist, found out something, which was already briefly mentioned in the previous chapter. *Jean Baptiste Joseph Fourier* said that any waveform can be generated as a sum of different sinusoidal oscillations. Smallest changes of added partial oscillations can contribute to a fundamental change in the waveform. The disadvantage that exists with additive sound synthesis, however, is the fact that increasingly complex sounds also require considerably more sinusoidal oscillations, which can degenerate into a large and almost unmanageable amount of work. Only by using computers with the provision of a more or less high number of oscillators can a satisfactory result be achieved, because in addition to an oscillator, components such as VCAs, ADSRs or various VCFs are needed. Let's take a closer look at the basics of additive sound synthesis.

The sine wave

The oscillator used in this chapter is called - who would believe it - VCO and looks like this. It has the known four outputs of the basic waveforms.



Abbildung 1 The VCO from Bogaudio

Now to the actual topic. Here again the sine wave with its frequency spectrum. In this case the VCO is set to a frequency of 1000Hz and in the spectrum at the position of 1K (1KHz=1000Hz) a deflection - called *peak* for short - can be seen.



VCV-Rack-Patch

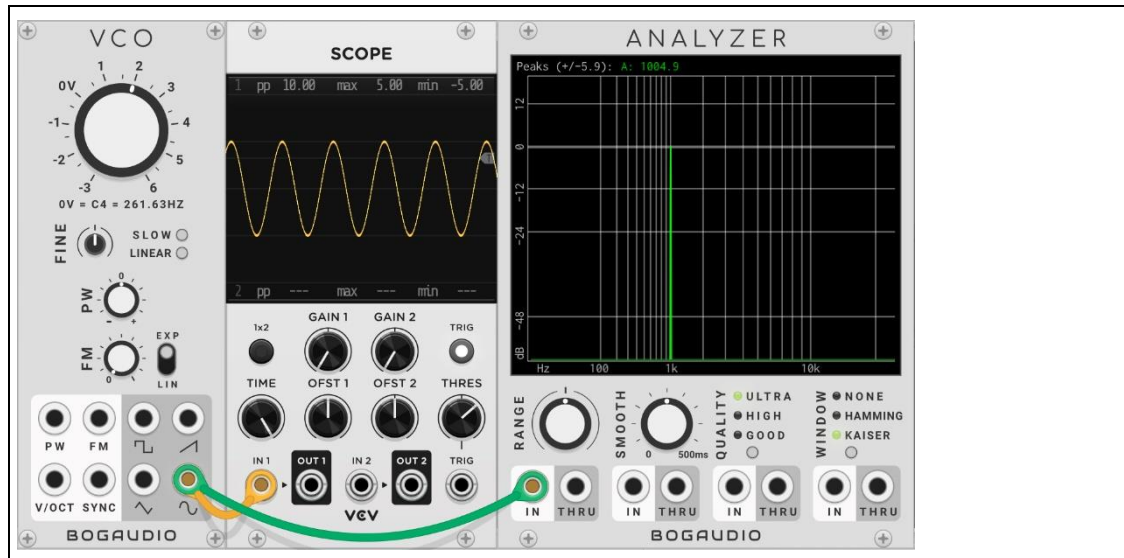


Abbildung 2 The sine wave with the frequency spectrum

So if we take this frequency spectrum as the starting point for sound synthesis, we would simply have to add a few more harmonics to the right of it, and the result would be new waveforms. Of course, these harmonics must be placed at very specific points in the frequency range, and the respective amplitudes should also have a certain magnitude and be in a certain relationship to each other. This is where the so-called harmonics come into play.



What are harmonics?

In technology, a harmonic is a harmonic oscillation whose frequency is an integer multiple of a certain fundamental frequency, whereby a harmonic above the fundamental frequency is also called a harmonic or overtone.

The 1st harmonic is equal to the frequency f of the fundamental, the 2nd harmonic has twice the frequency $2f$ and so on and so forth. On the following figure some oscillations can be seen.

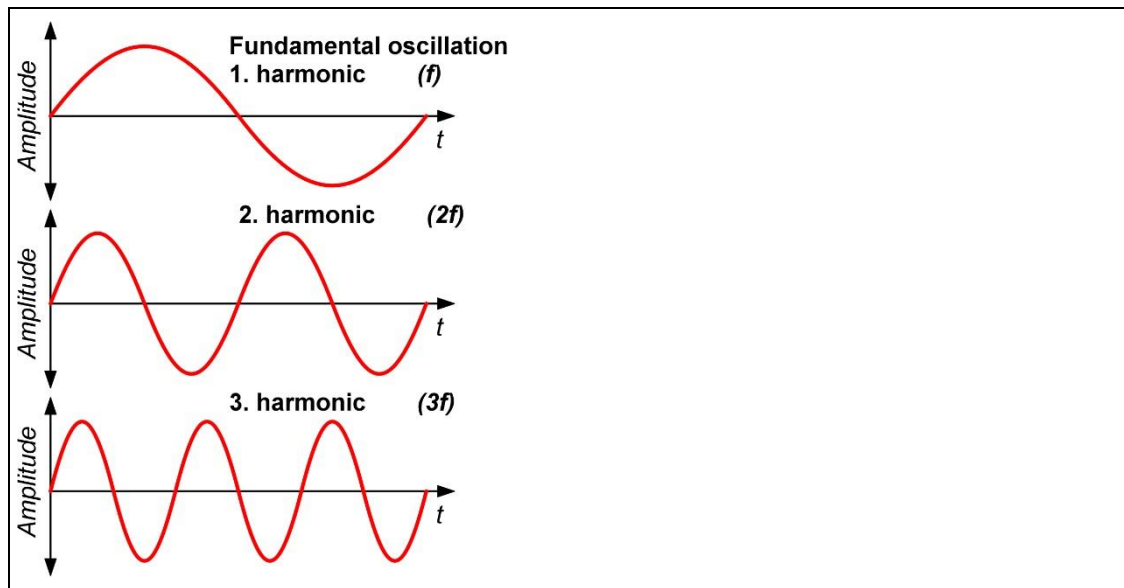


Abbildung 3 Harmonics

Each of the waveforms to be created is based on a certain generation pattern, taking into account the following parameters.

- Overtone spectrum (harmonics)
- Amplitude damping

The sound synthesis for a square wave signal

As an example for an additive sound synthesis a square wave shall be created. Now the fundamental frequency or the 1st harmonic must be enriched with further harmonics. But how do we get there without endless trial and error driving us mad? Of course there is a mathematically founded solution, whose derivation shall not be in the focus here. A square wave is generated via additive sound synthesis by adding an infinite number of sine waves of harmonics 1, 3, 5, 7, 9, etc. with certain amplitudes, but this is firstly not necessary and secondly impossible to realize. Is it possible to reproduce this with the VCV rack? Stupid question! Of course it is possible and we will have a closer look after the following formula. First it should be clear how the individual harmonics overlap and what has to be taken into account. The formula shows us that the frequency spectrum of a symmetrical square wave signal is composed of odd harmonics, where amplitudes of the harmonics decrease with increasing frequency and the divisor corresponds to the respective harmonic. Here the formula letter A stands for a certain amplitude and the formula for the circular frequency ω is $\omega = 2\pi f$.

$$f(t) = A \left[\sin(\omega t) + \frac{1}{3} \sin(3\omega t) + \frac{1}{5} \sin(5\omega t) + \frac{1}{7} \sin(7\omega t) + \dots \right]$$

The generation pattern is as follows for a square wave signal: (n = index of harmonics)

- Overtone spectrum: all odd harmonics
- Amplitude damping: $1/n$

Finally, the four sine waves and the resulting waveform look like this.

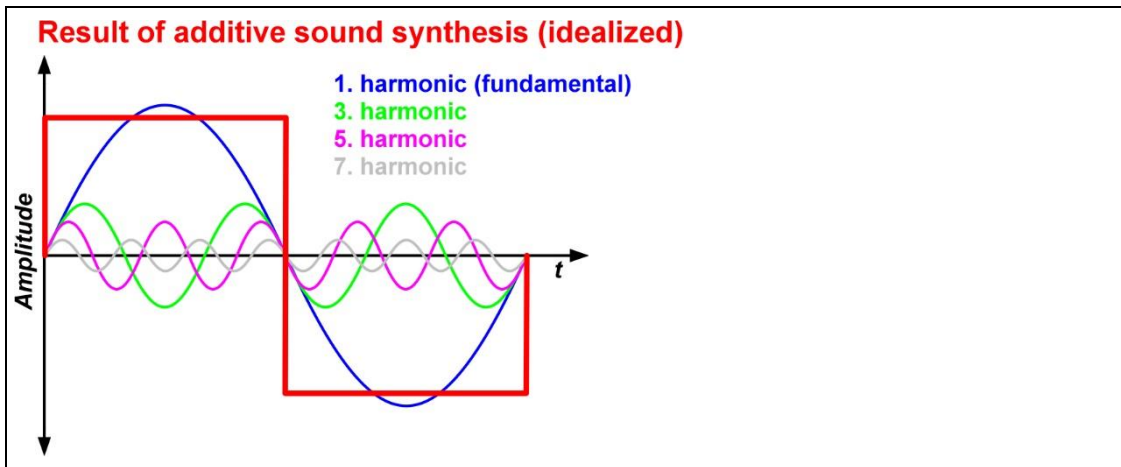


Abbildung 4 The additive sound synthesis with four sine waves

So we need four VCO modules that send their sine oscillations to the mixer module, for example, and of course we still need the scope module and a spectrum analyzer. However, there is one crucial point to keep in mind when we take another look at the four sine oscillations.



A small decisive hint on the side!

All oscillators used must be synchronized with the first oscillator, which provides the fundamental (1st harmonic).

Each oscillator has its own frequency and starts its oscillation period or phase independently of the other oscillators. If they didn't all start at the same time, we could stop using additive sound synthesis, because it would be a wild mess of oscillators oscillating somehow with different phase shifts. Before we move on to the VCV rack patch, let's take a look at the square wave signal in the spectrum analyzer and see if I haven't said anything stupid about the harmonics and amplitudes. The following figure shows only the range that is important for us, which normally extends much further to the right. The fundamental frequency of the first oscillator was set to 100Hz.

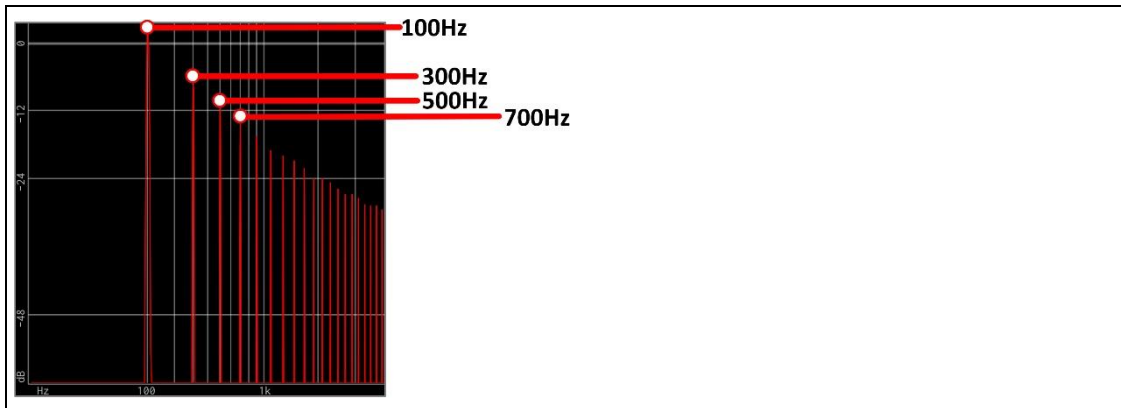


Abbildung 5 The section of a rectangular signal spectrum

It is wonderful to see that all harmonics correspond to the specifications of the formula shown, bearing in mind that this is a logarithmic scaling and the attenuation is given in dB. Now let's get down to business and set up the patch with the four oscillators and give them the above frequencies.



VCV-Rack-Patch



Abbildung 6 The formation of a square wave signal

In the Scope module, you can see a waveform that suggests a square wave signal. What is wrong here? Correct! The amplitudes have not yet been adjusted in the VCA MIX module, as the formula dictates. Now there is a small problem here, because the sliders of the individual input channels cannot be used to specify ratios such as 1/2, 1/3 or 1/4 of the signal strength. In this case, the information must be given in decibels. *Decibels*? That may sound like something you've somehow heard before and yet not quite understood! I do not take myself out there.



What means decibel?

When it comes to loudness or a noise level, the term *decibel* is used again and again, which is abbreviated with the two letters *dB*, whereby the decibel scale is logarithmic. However, decibel is not a measure of an existing volume, but compares two values with each other. The unchanged signal is assigned 0dB, which means that there is neither an increase nor a decrease in the noise level. The theory says that with +6dB there is a doubling of the sound pressure, which does not necessarily correspond to a doubling of the volume. In contrast, with -6dB you would have a halving of the sound pressure.

Back to our plan to convert the ratios into corresponding dB values. But of course there is a fancy formula that is helpful for this and reads as follows.

$$A = 20 \cdot \log_{10}(\text{Ratio})$$

Assuming a ratio of 1/2, i.e. 50%, we now have to determine the corresponding value in decibels. This would look as follows.

$$20 \cdot \log_{10}\left(\frac{1}{2}\right) = -6.02dB$$

So an attenuation of a signal by 50% corresponds to an attenuation by -6dB. In this way we can now convert the ratio values for a square waveform into dB.

Ratio	dB
1/1	0dB
1/3	-9,54dB
1/5	-13,98dB
1/7	-16,9dB

I now entered these values into the channel attenuation of the individual mixer channels and the waveform already looks very much like a square wave signal. The more oscillators would be used, the less the interfering waves would appear.



VCV-Rack-Patch

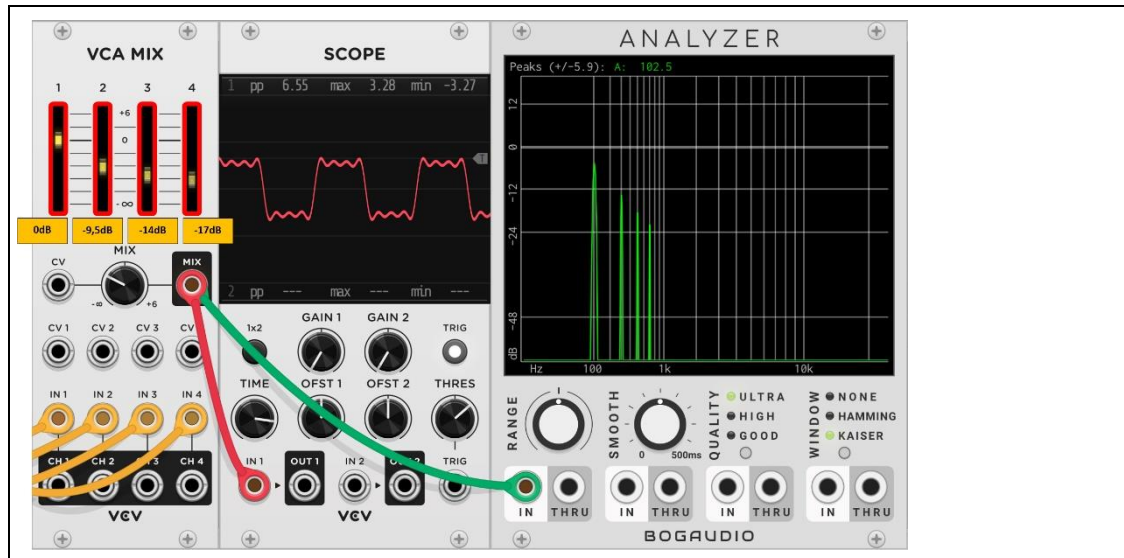


Abbildung 7 The corrected amplitudes of the four sinusoidal oscillations

Also in the spectrum analyzer you can see that the amplitudes of the harmonics have now adjusted according to the decibel values. On the following website, in addition to the VCV Rack, you can experiment a bit with additive sound synthesis and see how the waveform changes.



Hyperlink!

<https://meettechniek.info/additional/additive-synthesis.html>

There you will find a waveform generator that can be used to simulate what I just told you with multiple input signals. On the following figure you can see that I made a rudimentary setting for the odd harmonics.

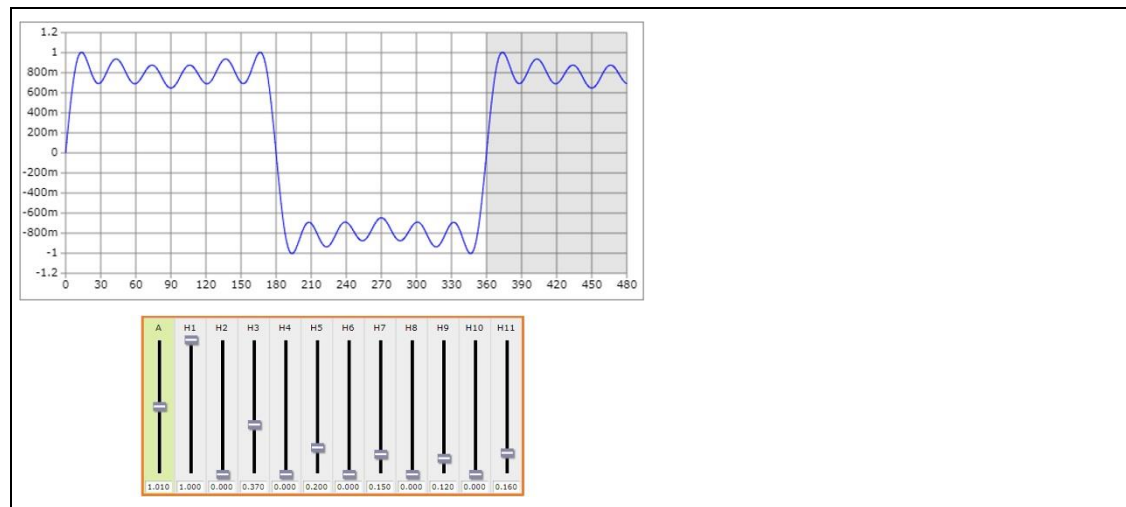


Abbildung 8 The additive synthesis with the waveform generator for a square wave signal

By using the sliders H1 (fundamental), H3, H5, H7, H9 and H11 I have synthesized a square wave - well not quite. I advise playing a little with the amplitude values here to see how it affects the waveform. Let's look at another example of sound synthesis.

The sound synthesis for a sawtooth signal

The synthesis of a sawtooth signal is similar to the synthesis of a square wave signal. However, the sawtooth signal is composed of even and odd harmonics and their amplitude corresponds to the quotient of the harmonics. The 2nd harmonic has an amplitude half the amplitude of the fundamental, the amplitude of the 3rd harmonic is 1/3, that of the 4th is 1/4, and so on and so forth. Since there is both a rising and falling form of a sawtooth signal, there are also different formulas.

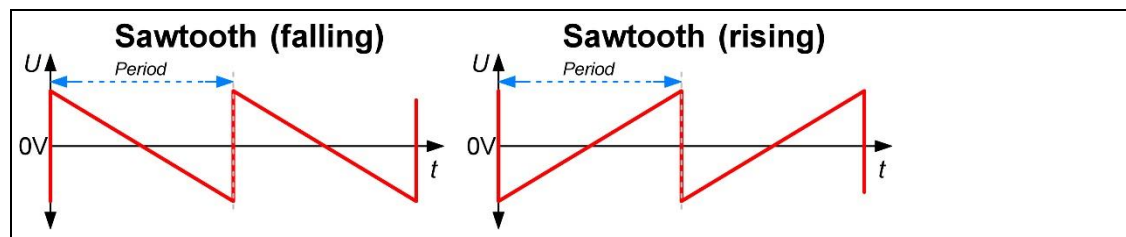


Abbildung 9 A rising and a falling sawtooth signal

The formula for the falling sawtooth signal is.

$$f(t) = A \left[\sin(\omega t) + \frac{1}{2} \sin(2\omega t) + \frac{1}{3} \sin(3\omega t) + \frac{1}{4} \sin(4\omega t) + \dots \right]$$

The formula for the rising sawtooth signal is as follows, with the even harmonics rotated in phase, i.e. inverted.

$$f(t) = A \left[\sin(\omega t) - \frac{1}{2} \sin(2\omega t) + \frac{1}{3} \sin(3\omega t) - \frac{1}{4} \sin(4\omega t) + \dots \right]$$

The generation pattern for a sawtooth signal is as follows: (n = index of harmonics)

- Harmonic spectrum: all harmonics
- Amplitude damping: $1/n$

Due to the relatively low amplitude attenuation, a lot of harmonics are audible, so that a full sound image is created. This time I used the Attenuator module *8VERT* to adjust the amplitudes with the help of the ratio. Then the conversion via the ratio-decibel formula is omitted. Let's have a look at the spectrum of a pure sawtooth signal before. The fundamental frequency of the first oscillator was again set to 100Hz.

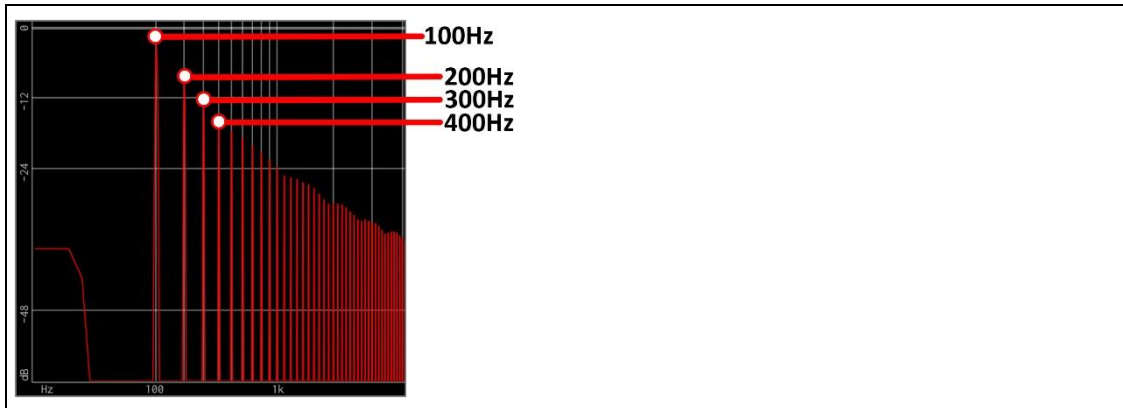


Abbildung 10 The section of a sawtooth signal spectrum

It is wonderful to see that all harmonics correspond to the specifications of the formula shown. The patch is as follows, whereby, as already mentioned, the *8VERT* module is now responsible for adjusting the individual amplitudes. The sliders of the mixer module must remain in the default position of 0dB for this. We remember: 0dB means no influence (attenuation or amplification) of the signal.



VCV-Rack-Patch

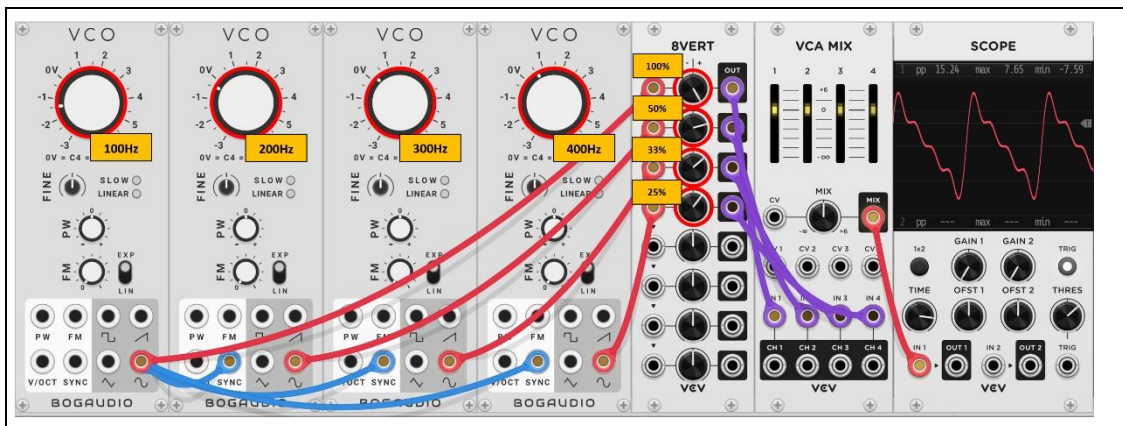


Abbildung 11 The formation of a sawtooth signal - falling

In the Scope module you can see a waveform that is very close to a sawtooth signal. A little brain teaser: Which two knobs on the *8VERT* module have to be set to which values to turn the falling sawtooth signal into a rising one?

The sound synthesis for a triangle signal

Let's conclude the additive sound synthesis with the formation of a triangular signal. We have now dealt with the four basic waveforms - sine, square, sawtooth and triangle. The triangle signal is very close to the sine signal both sonically and visually. It contains only a few odd harmonics whose volume decreases exponentially. The measure of the attenuation corresponds in each case to the square of the number of the harmonics. The formula for this is as follows, where the sign is negative for every second harmonic.

$$f(t) = A \left[\sin(\omega t) - \frac{1}{9} \sin(3\omega t) + \frac{1}{25} \sin(5\omega t) - \frac{1}{49} \sin(7\omega t) + \dots \right]$$

The generation pattern for a triangle signal is as follows: (n = index of harmonics)

- Overtone spectrum: all odd harmonics (every second negative)
- Amplitude damping: $1/n^2$

Let's have a look at the spectrum of a pure triangle signal before. The fundamental frequency of the first oscillator was again set to 100Hz.

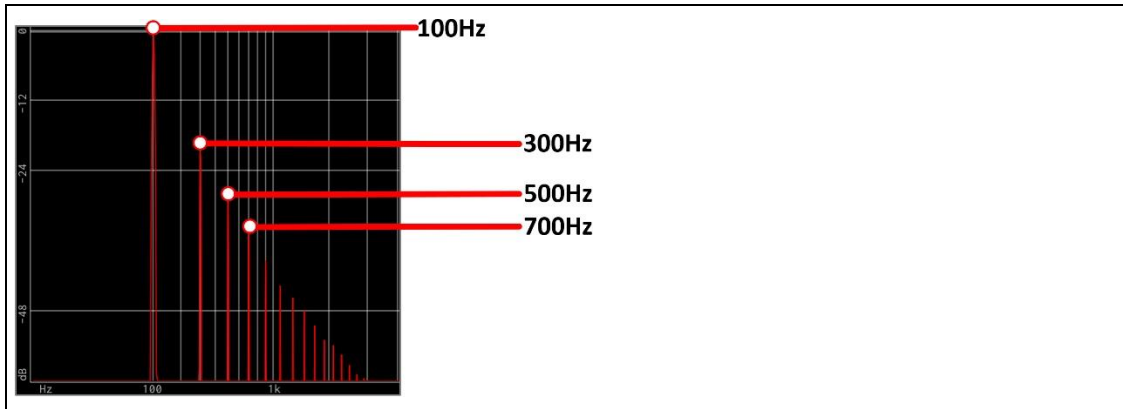


Abbildung 12 The section of a triangle signal spectrum

Again, it is easy to see that all the harmonics conform to the specifications of the formula shown, but the amplitude of the harmonics decreases very quickly. The corresponding patch looks like this.



VCV-Rack-Patch

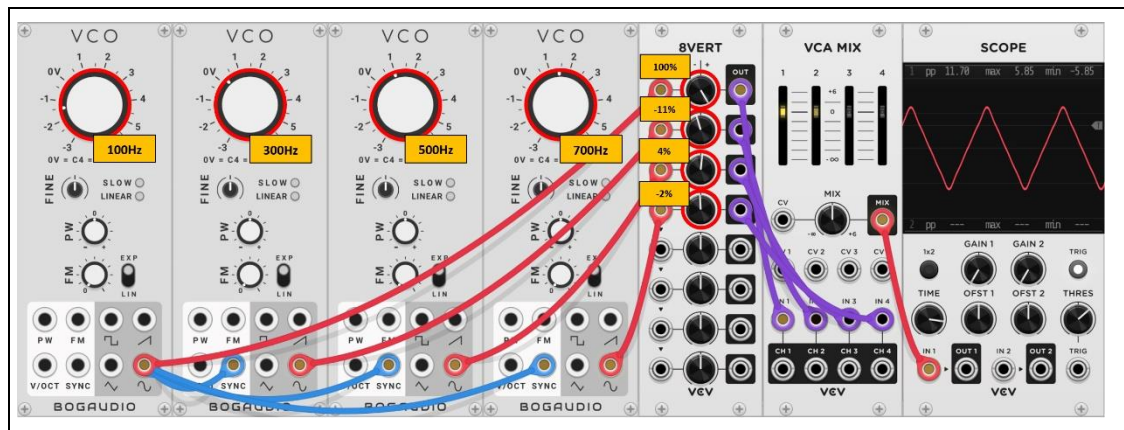


Abbildung 13 The formation of a triangular signal

In the Scope module, a waveform can be seen that closely resembles a triangular signal. Here again the 8VERT module was used for amplitude attenuation.

The subtractive sound synthesis

In subtractive sound synthesis, the fundamental is generated by a simple waveform that has as many harmonics as possible, as is the case with a square wave or sawtooth, for example. If a lot of harmonics are present, it is then very easy to filter out something here and there from the frequency spectrum to achieve a desired sound result. In subtractive sound synthesis, a sine wave as the fundamental would make no sense, because there are no harmonics to filter out. Only the amplitude, i.e. the volume, would be adjusted. Low-pass, high-pass or band-pass filters are used as filters here, to name just a few. Since harmonics, as the name suggests, are located above a fundamental, the use of a low-pass filter is very suitable, which filters out more or less high frequencies from the harmonic spectrum with a variable cutoff frequency. But the sound result, or rather the sound experience, as a current end product would be something like carved in stone and therefore very static and lifeless. But life has something to do with movement and for this reason the already mentioned modulation comes into play, as for example an LFO can do very well with a rhythmic shift of the cutoff frequency.



VCV-Rack-Patch



Abbildung 14 The subtractive sound synthesis

In this patch you can see that the LFO module modulates the cutoff frequency of the VCF module and more or less harmonics can be seen in the spectrum analyzer. The sound changes, which can also be seen in the waveform in the Scope module.

Other sound synthesis

However, additive and subtractive sound synthesis are not the only ways to create sounds. Other synthesis principles are for example

- The wavetable synthesis - Comes up in this chapter
- The FM synthesis - Will be discussed in more detail in the FM chapter
- Cross-Modulation - Was addressed in the previous Oscillator chapter
- Pulse Width Modulation - Was addressed in the previous oscillator chapter
- Granular synthesis

Of course there are many more principles, but to list and explain them all would go beyond the scope of this article and would be too much for the moment. As another example I show the wavetable synthesis, which is used very often.

The wavetable synthesis

Why is the term waveform table used? What a table is, should be clear to everyone. It is a list-like compilation of certain information. Since some synthesizers are able to store different waveforms in order to recall them when needed, very interesting effects can be achieved by this. Of course, it then makes little sense to always load a single waveform to play it. It only becomes exciting when several different waveforms are stored in the aforementioned *wavetable* and these are called up and played one after the other in a certain way - for example via an LFO or an envelope. This can create a very lively sound, which is ideal for pads and strings in particular.



What is a pad sound?

A pad sound is a more or less sustained tone or chord produced by a synthesizer to create a kind of background harmony with a warm character over it. Especially when it comes to generating atmospheric sounds, such as the strings do in an orchestra, pad sounds are very suitable. With respect to the envelope and the attack, sustain and release parameters, these are set to allow for a smooth attack or release, which is achieved via correspondingly high values. This counteracts a strong level change and enables a smooth surface sound. If a second oscillator is now added, which is slightly out of tune with the first, a *levitation* is achieved which further increases the effect. A *levitation* is an effect obtained by superimposing two oscillations that differ only very slightly in their frequencies. This leads to a periodic fluctuation - i.e. an increase or decrease - in amplitude.

A changing waveform can be stored in a wavetable and is then available for recall. The following figure shows such a wavetable with six waveforms.

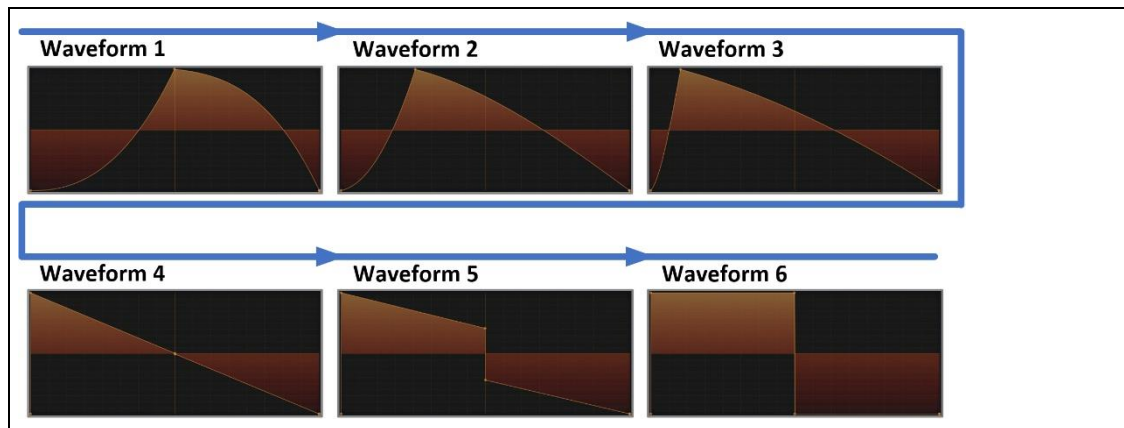


Abbildung 15 The waveforms of a wavetable

There is a transition from *FIN* to *SAW* to *PULSE* here. If this wavetable is played, an internal morphing between the existing waveforms takes place. To experiment with a wavetable module, the *Black Wavetable VCO* from *EricaSynths* is certainly worth a look.



Abbildung 16 The Black Wavetable VCO from EricaSynths

At the following internet address you can find the modules of EricaSynths as a plugin.



Hyperlink!

<https://library.vcvrack.com/EricaCopies>

Below are two patches that control the wavetable module. In the first one it is the LFO and in the second one an ADSR envelope that calls up the wavetables.

The control by the LFO module

If the wavetable module is modulated by an LFO, the stored waveforms are called up cyclically when a key on the keyboard is pressed.



VCV-Rack-Patch

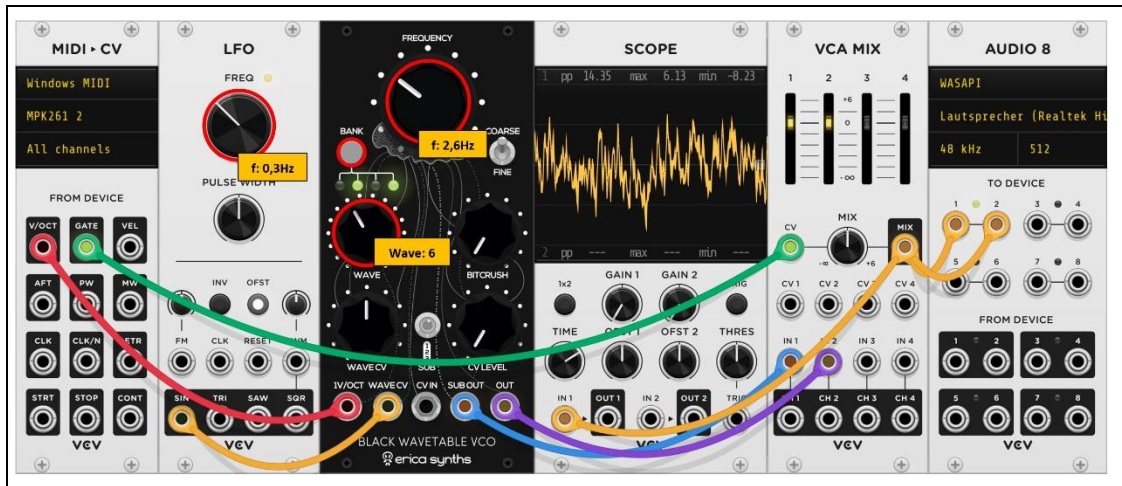


Abbildung 17 An LFO controls the wavetable VCO

The control by the ADSR module

If the wavetable module is modulated by an envelope - when a key on the keyboard is pressed - the stored waveforms are called up once, but this depends on the set CV level.



VCV-Rack-Patch

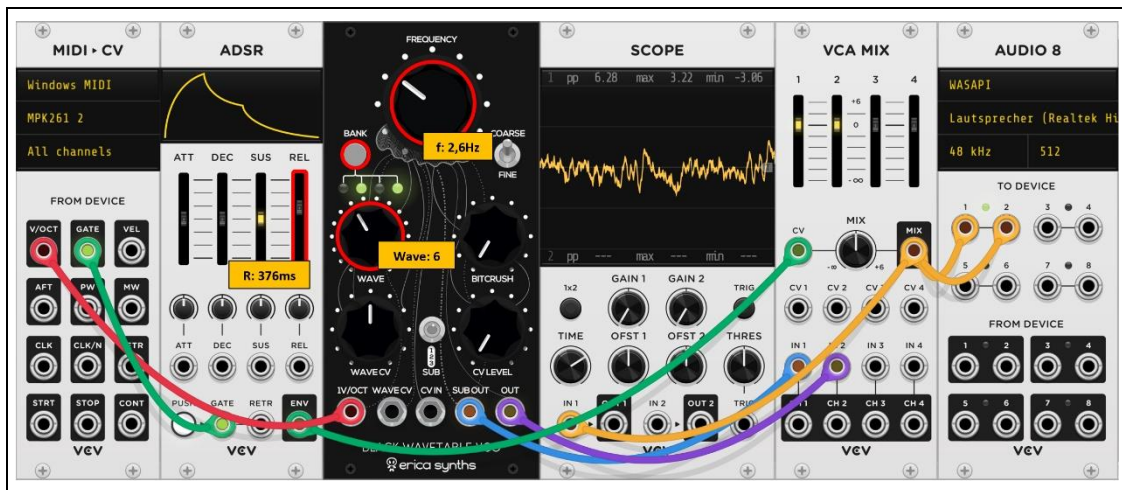


Abbildung 18 An ADSR envelope controls the wavetable VCO

By adjusting various parameters, this should again be experimented with to see and hear what effect this has.

The VCV-Rack - Hardware control



What is it all about?

The following topics are discussed in this chapter.

- The control of hardware
- The CV-to-MIDI-Module
- The control of a real synthesizer
- MIDI or USB?
- Recording an audio stream from an external audio interface
- The CV-MIDI-CC-Module
- What are continuous controller signals?
- What is an envelope follower?
- The Follow module from Bogaudio

VCV-Rack is, after all, a software that is self-contained and all modules are virtual in nature. In the chapter that dealt with MIDI mapping, you could see how external hardware in the form of a MIDI controller could directly influence certain parameters of modules. Now it would certainly be quite exciting if it were possible to influence or control connected hardware via VCV-Rack. There are different approaches in this respect and I would like to start this chapter with one of them.

The CV-to-MIDI module

Now the MIDI-to-CV module has been used several times when it came to making the connected master keyboard usable as an interface between human and VCV-Rack. But there is also the exact opposite way in VCV-Rack, which sends MIDI information via a certain module. It is called *CV-to-MIDI* module.

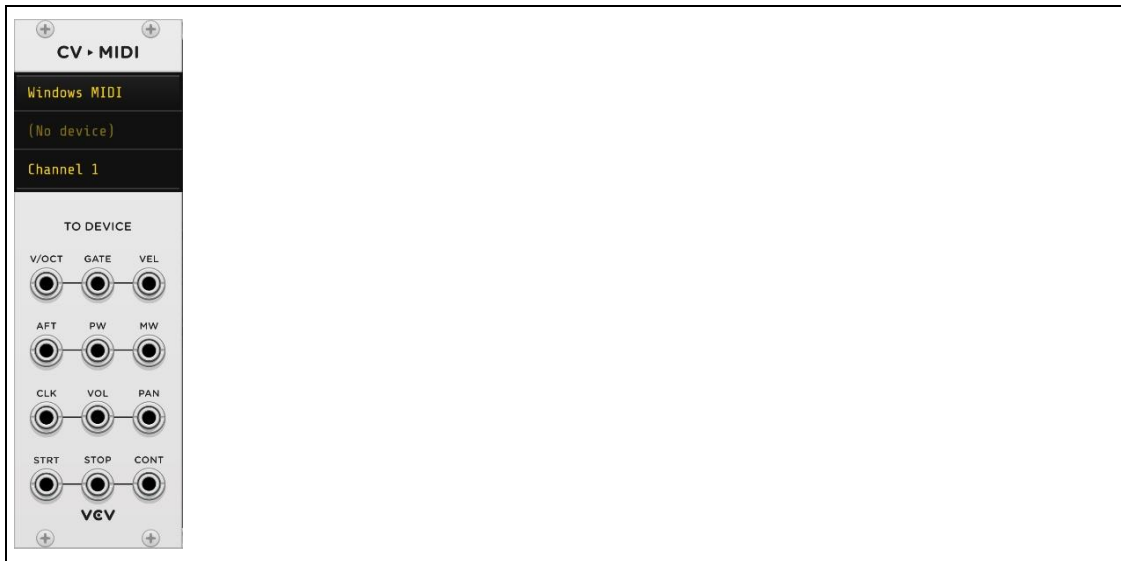


Abbildung 1 The CV-to-MIDI-Module

But what is it all about? I'm going to have to explain it a little bit. The topic MIDI has already been discussed a bit more in the MIDI Mapping chapter. Now I would like to expand this a little bit. Since in a music studio surely different hardware communicates over the MIDI interface, it is necessary to address these individual devices in some way unambiguously. In the transferred sense it behaves similarly as in a network, in which several participants are and with each other and among themselves information exchange. For this purpose, it is imperative that each participant has a unique address, because otherwise it is impossible to uniquely identify a particular addressee. The where from and where to must therefore be clearly defined. The situation is quite similar with MIDI, even if in a somewhat reduced form. The already mentioned original MIDI standard defined 16 channels. So for the real world this can mean, for example, a street with 16 buildings, which have house numbers from 1 to 16. If a letter carrier passes by, he can identify a certain house exactly by these numbers. So each sound generator in the MIDI network is assigned a unique number called a MIDI channel. A master keyboard is now used to determine which MIDI channel is to be selected. Each MIDI sound generator that is to be controlled must therefore be assigned its own channel. However, this does not necessarily have to be unique, because there are also special cases where several sound generators have the same MIDI address and play simultaneously when controlled. The so-called MIDI multimode should not play a role here. Let us come to a concrete scenario.

The control of a real synthesizer

The following example can be done with all kinds of real sound generators and I use my synthesizer *MatrixBrute* for it. The VCV rack works in such a mode as a so-called *HUB*, i.e. as a node to receive, process and then pass on data.



Abbildung 2 The Synthesizer MatrixBrute

Now I want to control this synthesizer via VCV-Rack. For this, either a MIDI connection or a USB connection to the computer must be established. The MIDI connection is usually made via the connected audio interface, which is provided with corresponding MIDI jacks.

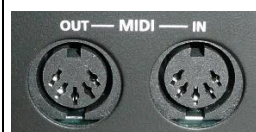


Abbildung 3 MIDI jacks

But I prefer to use the USB variant, which can also send MIDI signals via the USB protocol. On the following picture both variants are once clarified.

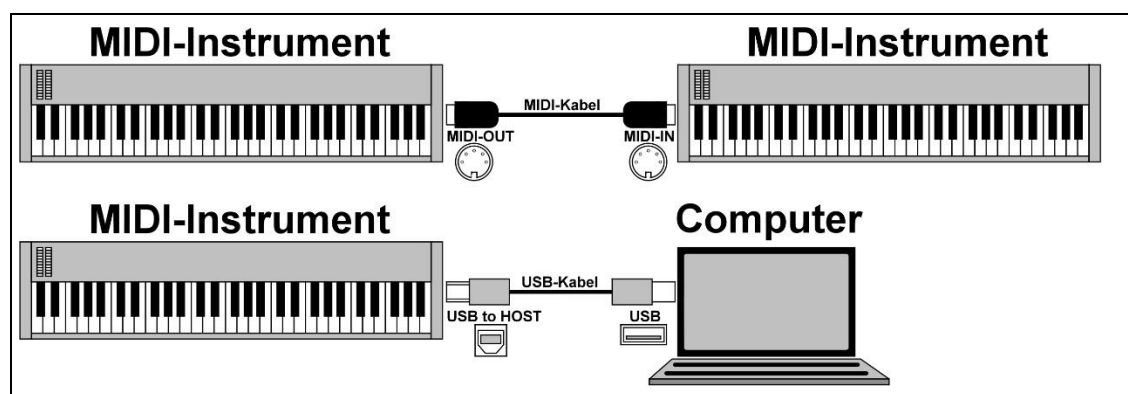


Abbildung 4 Various MIDI connections

After connecting to my computer, the setup of the synthesizer can be opened to set the MIDI channel. I show this here exemplarily for the MatrixBrute, what however each other synthesizer surely offers in similar form. You can see that I have set the input channel for MIDI to 7. We then have to set this channel in VCV-Rack right away.

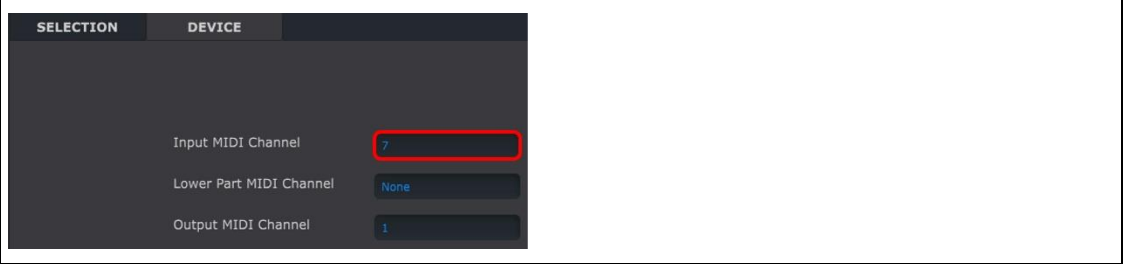


Abbildung 5 MIDI-Setup for the MatrixBrute

Now let's take a look at the CV-to-MIDI module.

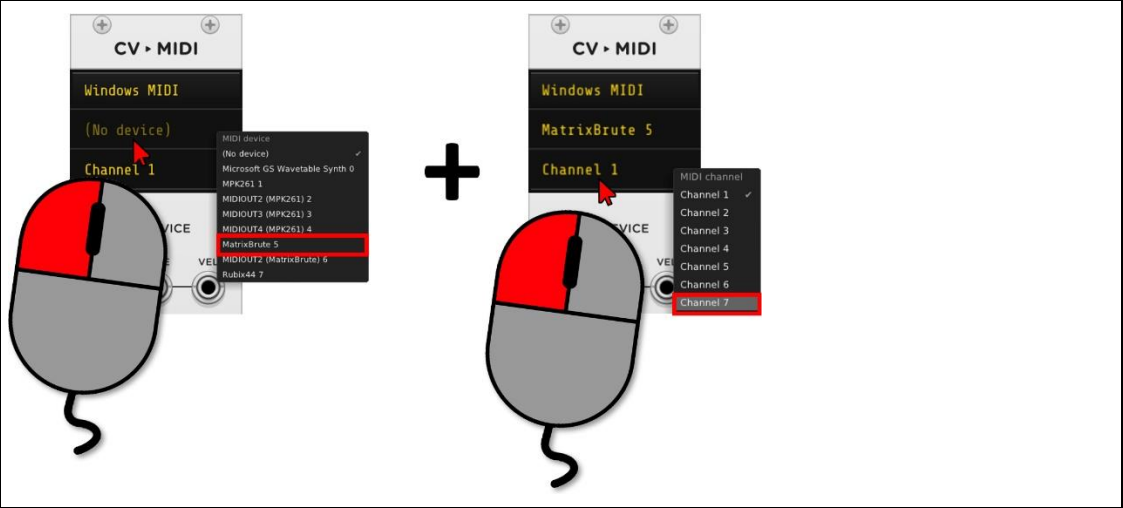


Abbildung 6 The MatrixBrute entry in the CV-to-MIDI module

After the selection the corresponding entry is in the module and this is then configured with the correct MIDI channel 7. The number 5 behind the device description is only an index, which is used in the list of recognized devices and has no further meaning.



Abbildung 7 The MatrixBrute entry is selected and configured in the CV-to-MIDI module

A first test

A first test, if everything arrives also in the direction of the MatrixBrute regarding MIDI, can be done via the following two modules.



VCV-Rack-Patch

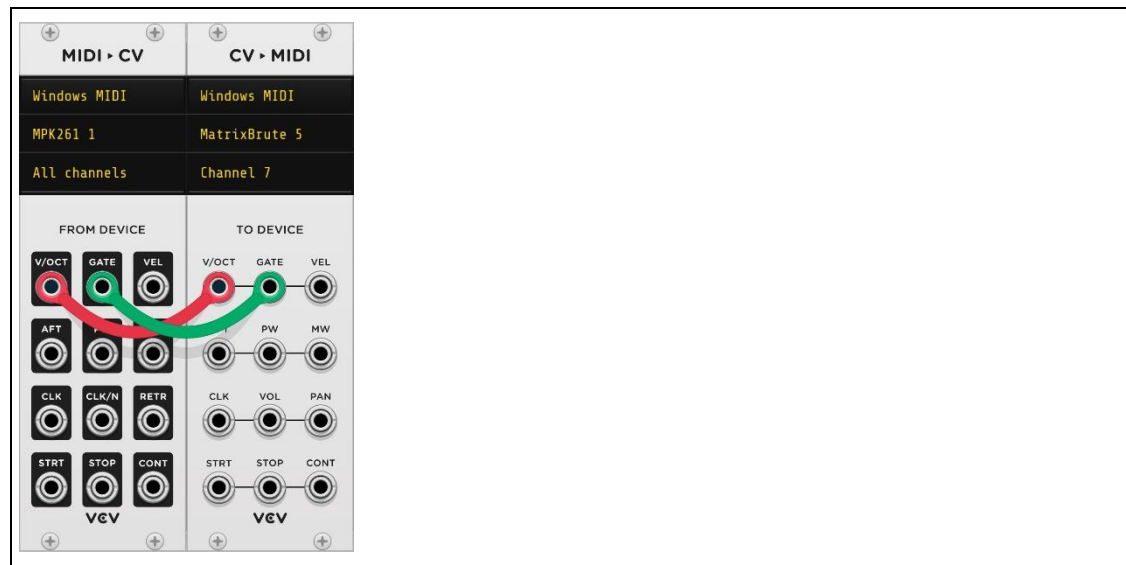


Abbildung 8 The patch for testing the MIDI connection to the MatrixBrute

When I play my keyboard now, the MatrixBrute reacts as an external synthesizer and plays the corresponding notes. However, the sound output is not done in the previous way, because there is no audio module. The MatrixBrute is connected to the external audio interface and there are so-called monitor boxes on it. Is this formulated understandably? I think that a block diagram can certainly do that better.

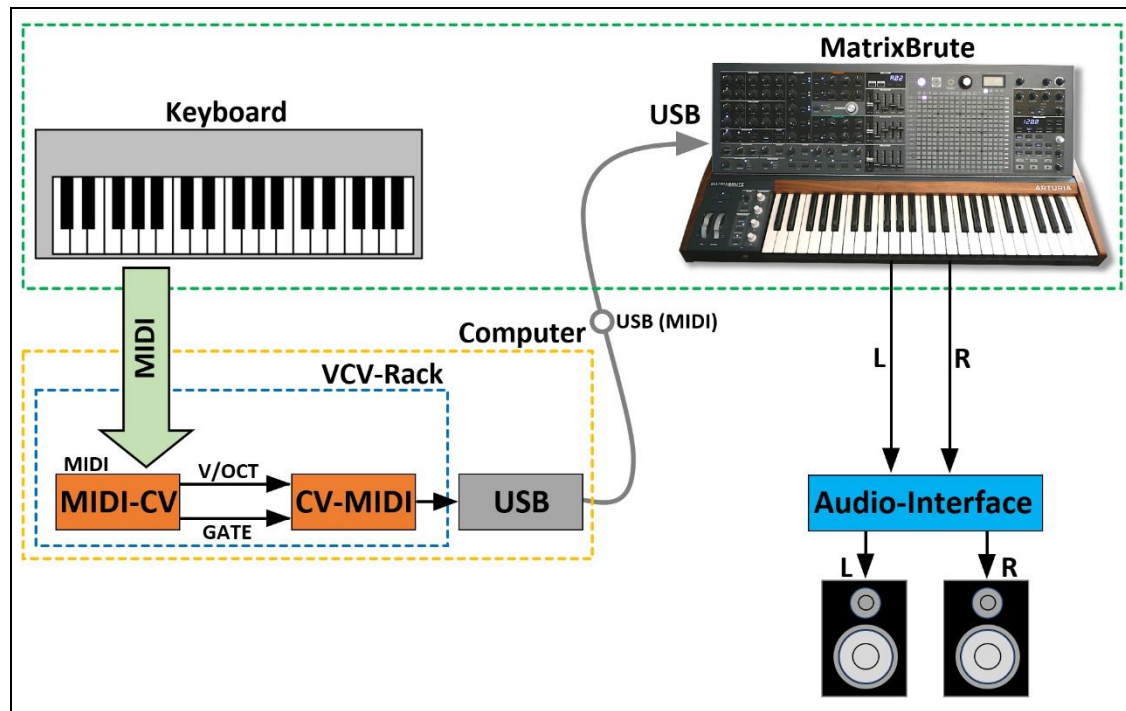


Abbildung 9 The block diagram for the communication between VCV rack and MatrixBrute

At first glance, it doesn't make much sense to use the master keyboard to control the MatrixBrute, which has its own keyboard. In the first step I only wanted to demonstrate the communication between the VCV rack and an external MIDI device.

A second test

Maybe some of you have the wish to somehow get the sounds of the external synthesizer or another connected MIDI device into the environment of the VCV rack. Haven't we had this scenario before? That's right, because the *REC* module did offer the ability to record an audio stream into an audio file. But after all, the Matrix Brute does not route its audio signals to the computer, but to the audio interface. But how is it then possible to tap this stream and send it to the REC module so that it can be saved? Since the external audio interface is also a USB device and connected to the computer! So it is possible to select this in the Audio 8 module, for example, and then connect the output to the REC module. No sooner said than done!



VCV-Rack-Patch

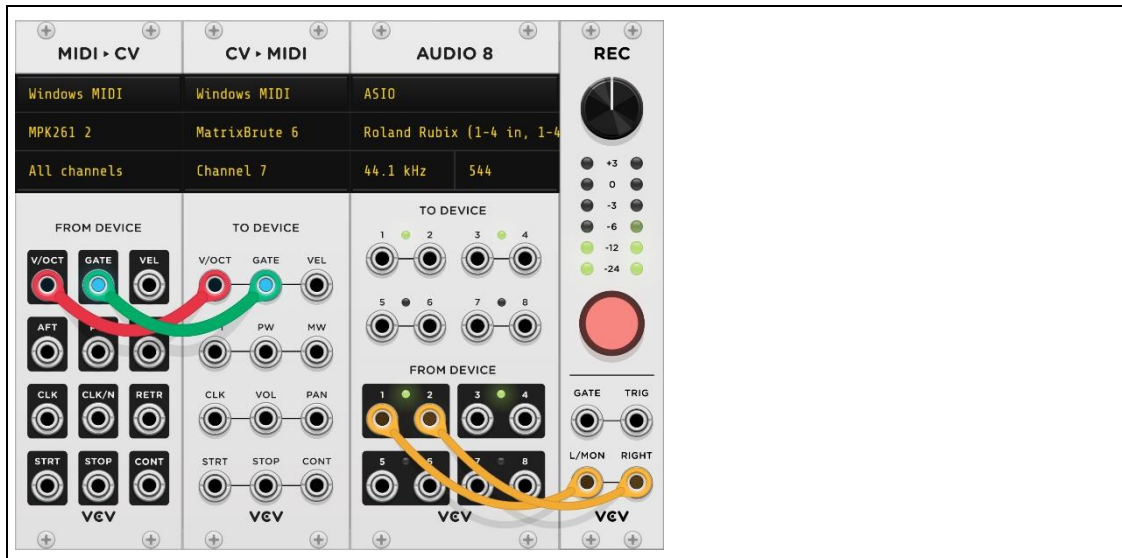


Abbildung 10 The REC module records the audio signal from the MatrixBrute and stores it in the file system

The saved audio file can then be used for a variety of purposes in numerous modules such as WT-VCO or sampler.

A complex sequencer control

Now the MatrixBrute already has a sequencer and arpeggiator internally, but with this patch I want to show how such a control can be done via VCV Rack. For this I use again the SEQ-3 module with downstream quantizer. So far this is nothing new. But I would like to expand the whole thing a bit, because a hardware synthesizer usually also has so-called *MIDI Continuous Controller* assignments. Sounds pompous, but it is quite simple. Until now, the CV-to-MIDI module was used to send specific MIDI notes via the dedicated MIDI channel. The hardware synthesizer reacted with different tones. But now there is also the possibility to access and modulate the different parameters of a synthesizer. We know this from the already known VCV rack modules, where at a VCO the pitch or at a VCF the cutoff frequency was modulated. Such parameters can also be sent via a certain VCV-Rack module, so that the hardware synthesizer reacts to them. What follows now is only an example for the Matrix-Brute and can basically be applied to any other synthesizer. It is only necessary to consult the respective manual. But let's start with the VCV rack module that holds this functionality. It is called *CV-MIDI-CC*. So CV signals are transferred via MIDI into CC signals (continuous controller).

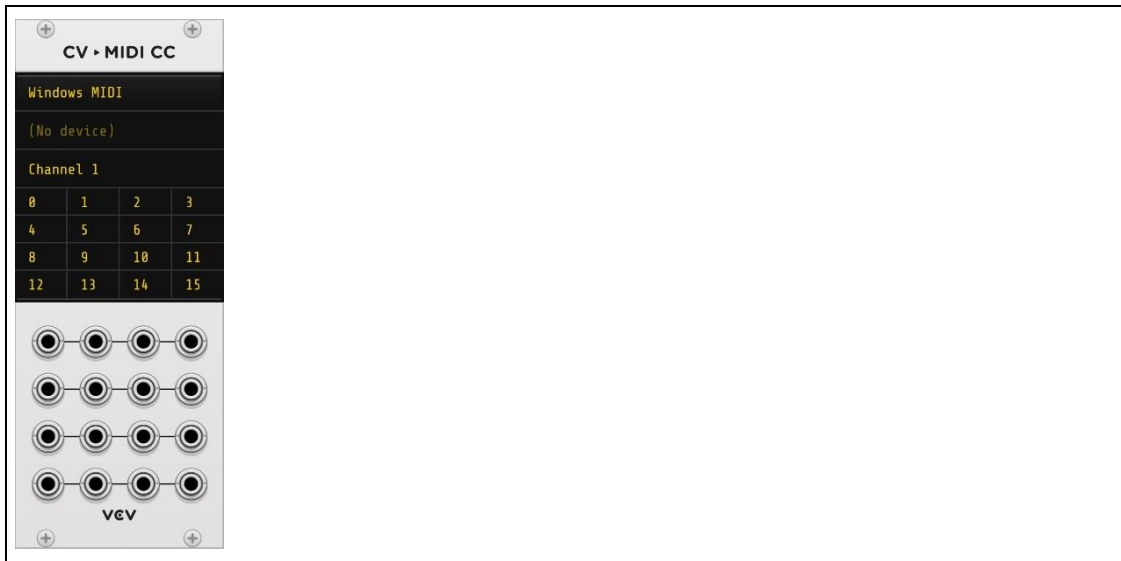


Abbildung 11 The CV-MIDI-CC-Module

I spare myself the details about the configuration of Device and Channel, because it is the same procedure as with the CV-to-MIDI module. Decisive here is the lower part, which shows the so-called MIDI-CC numbers. At the moment these are numbered from top left to bottom right from 0 to 15, but can be adjusted accordingly with a mouse click. Now to send a CV signal to a selected number, you have to use the corresponding socket, which is located below the number matrix. For my MatrixBrute there are the following assignments regarding the *VCO-1*, just to show a small introduction. The list is somewhat shortened, but this does not matter.

VCO-1-Parameter	MIDI CC	Resolution
Fine (Pitch-Finetuning)	65	7-bit
Coarse (Pitch-Tuning)	66	7-bit
Waveform Selection	67	7-bit
etc.		

So to adjust the pitch via CC, the MIDI CC number 66 must be sent with a value between 0 and 127. However, I don't want to influence the VCO in any way, but the master cutoff of the MatrixBrute. Here is the corresponding parameter.

Filter Parameter	MIDI CC	Resolution
Master Cutoff	27	7-bit/14-bit

So the number 27 is the correct one here, which must be provided with a certain value. The following patch shows the implementation.



VCV-Rack-Patch



Abbildung 12 The patch controls the pitch and cutoff of the MatrixBrute

The Envelope follower

Now let's move on to another topic that deals with audio signals coming into VCV Rack from the outside to do something there. How is it possible to use an audio signal to modulate a certain parameter, for example? If you take a closer look at an audio signal, you will notice that it looks very diffuse when displayed in a scope. How should it be possible to pick out a certain value in order to derive a control signal from it?

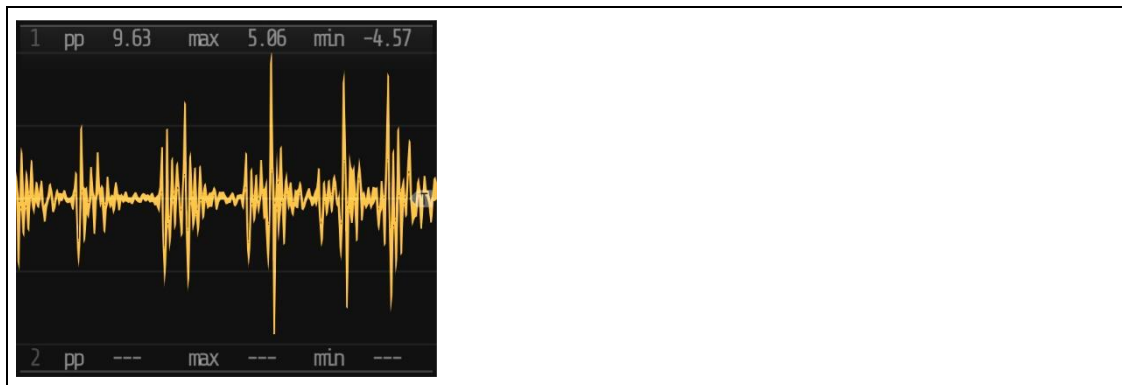


Abbildung 13 Any audio signal

In terms of time, the jumps in amplitude are already very hard, so that it would be just as hard here with a direct conversion to a CV signal. That would not be what you want in the end. There would have to be

a way to ride on top of the peaks, so to speak, without falling into a following valley in each case. Of course there is a solution for this aspect in the form of a so-called *envelope follower*.



What is an envelope follower?

An envelope follower is then used to convert the volume curve of an audio signal into a control signal. The module continuously samples an audio signal and generates a continuous control voltage from the determined volume curve.

I think it's best to take a closer look at this first on a test patch, which looks like this. The *FOLLOW* module from Bogaudio is used here, which can be seen in the following picture. The sensitivity or gain can be adjusted individually via the rotary controls.



Abbildung 14 The *FOLLOW*-Module from Bogaudio

It is easy to see that the red curve, which is the output signal of the *FOLLOW* module, exactly follows the peaks of the yellow audio signal. This red curve can be used directly as a control signal for modulating certain parameters.



VCV-Rack-Patch

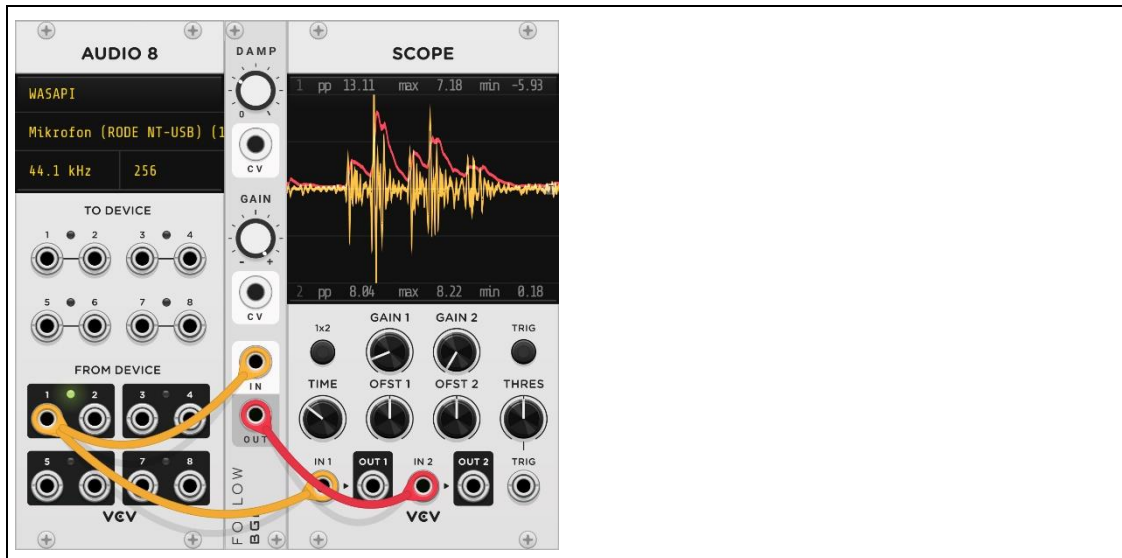


Abbildung 15 The envelope follower in use

With the following patch I only give a suggestion to show how the output signal of the envelope follower can be used to control one or more parameters. In my case the V/OCT of the FM-OP module is influenced by the connected microphone. The OUT signal of the FOLLOW module also drives the GATE input of the FM-OP module, which only sends a signal to its output if an audio signal is also present and a certain level is exceeded. In addition, the ENV button on the FM-OP module is pressed to activate the envelope control.



VCV-Rack-Patch

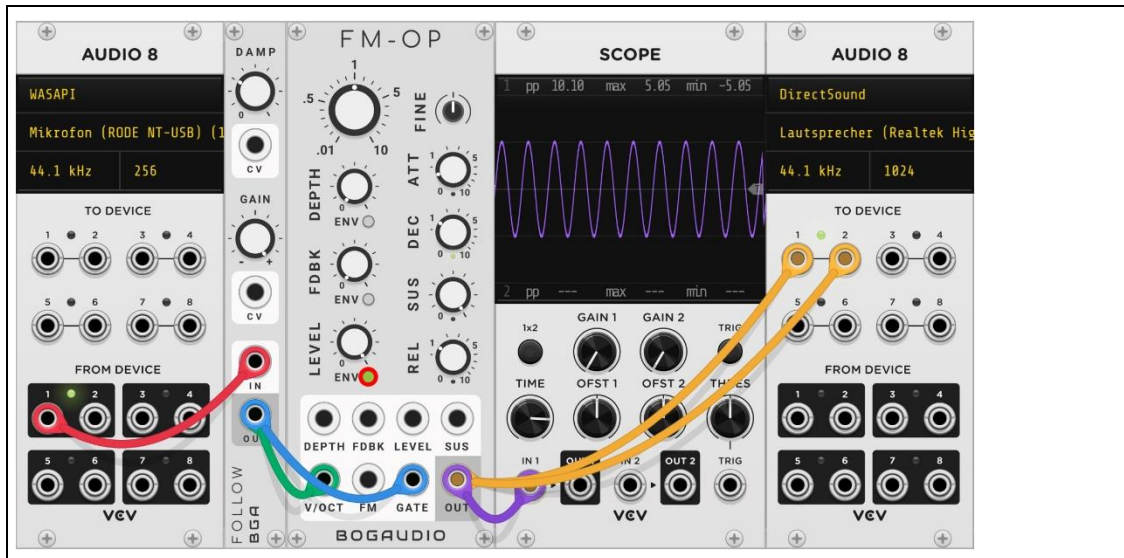


Abbildung 16 The envelope follower controls the pitch of the FM-OP module

This should give a foretaste of what is possible to control or modulate a connected external hardware as well.